Modeling Continuous Data Using Mixture Models

Daniel W. Heck

Statistical Modeling in Psychology
Freiburg Heidelberg Landau Mannheim Tübingen

2023-04-23

MPT Modeling with Continuous Data

1) MPT-RT: Modeling response times with histograms
   - Heck & Erdfelder (2016)
2) GPT (generalized processing tree): Parametric modeling
   - Heck, Erdfelder, & Kieslich (2018)
3) RT-MPT: Serial-process model for response times
   - Klauer & Kellen (2018)

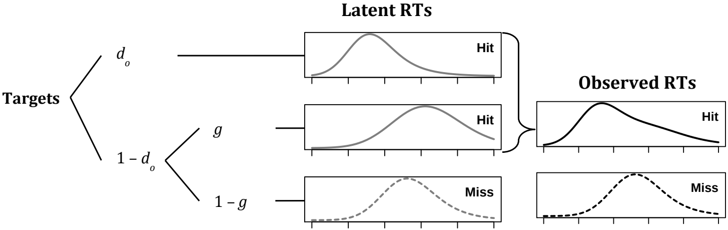
MPT Models and Continuous Variables

- Discrete-state modeling for discrete & continuous variables
  - Response times, confidence ratings
  - Process tracing measures (eye or mouse tracking)
  - Neurophysiological data (e.g., amplitudes of ERP signals)
- Data structure:
  - In each trial we observe one discrete response and one or more continuous values (e.g., response time)
  - In standard MPT modeling, we would simply ignore all continuous measures and make a frequency table of discrete responses

Item Type	Discrete Response	Response time
Target	"old"	930
Target	"new"	1532
Target	"old"	1240
...	...	
Lure	"old"	798
Lure	"new"	2332
...	...	

Mixture distribution

- Core idea: same MPT mixture distribution for discrete & continuous data
  - A) Latent RTs: Different processing branches of the MPT model result in different latent distributions $g_j(t)$
  - B) Observed RTs: A mixture distribution, defined as $f(t) = \sum_j p_j g_j(t)$
  - C) Mixture weights p_j are determined by the MPT branch probabilities

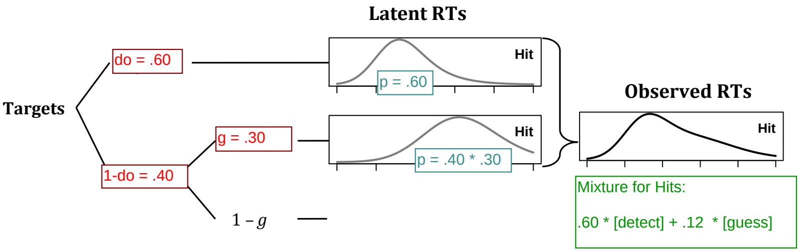


Example: Mixture Distribution

Illustration: 2-high threshold model

- Latent RTs:
  - $g_{\text{detect}}(t)$ = RT distribution for detection
  - $g_{\text{guess}}(t)$ = RT distribution for guessing
- Observed RTs for correct "old" responses to targets:

$$f(t, \text{Hit}) = d_o \cdot g_{\text{detect}}(t) + [(1 - d_o)g] \cdot g_{\text{guess}}(t)$$



Differences of the Approaches

Three different approaches

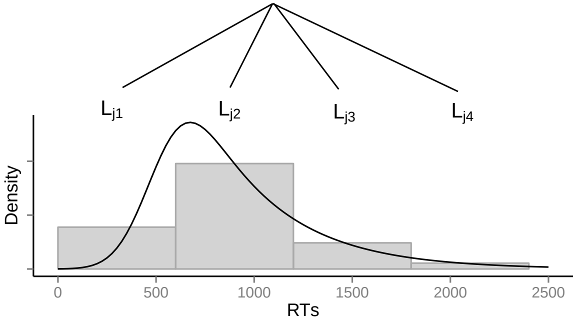
- The three methods *all* assume mixture distributions for continuous variables
- Main difference: Assumptions for the component distributions $g_j(t)$
  - A) Histogram/nonparametric
  - B) Any parametric distribution
  - C) Serial-processing assumptions

MPT-RT: Modeling response times with histograms

- Heck, D. W., & Erdfelder, E. (2016). Extending multinomial processing tree models to measure the relative speed of cognitive processes. *Psychonomic Bulletin & Review, 23*, 1440–1465.
- Heck, D. W., & Erdfelder, E. (2017). Linking process and measurement models of recognition-based decisions. *Psychological Review, 124*, 442–471.

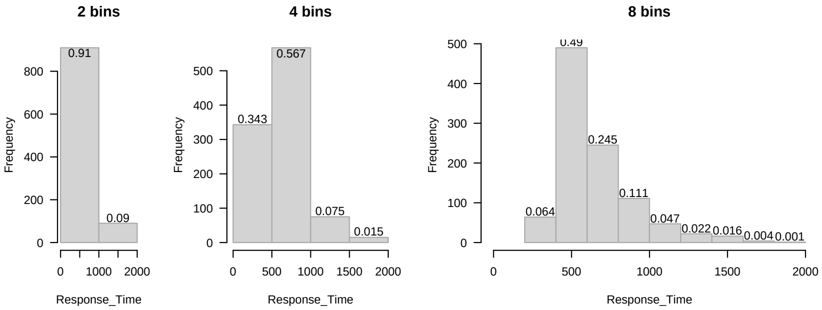
Histogram-Based Approach (Heck & Erdfelder, 2016)

- Categorize RTs into discrete bins (Yantis, Meyer, & Smith, 1991)
  - Example: "Very fast", "fast", "slow", "very slow"
- State-specific distributions are modeled by the parameters L_{jb}:
  - $L_{jb} =$ height of the histogram bins
  - $L_{jb} =$ probability that state j results in observation in the b-th interval



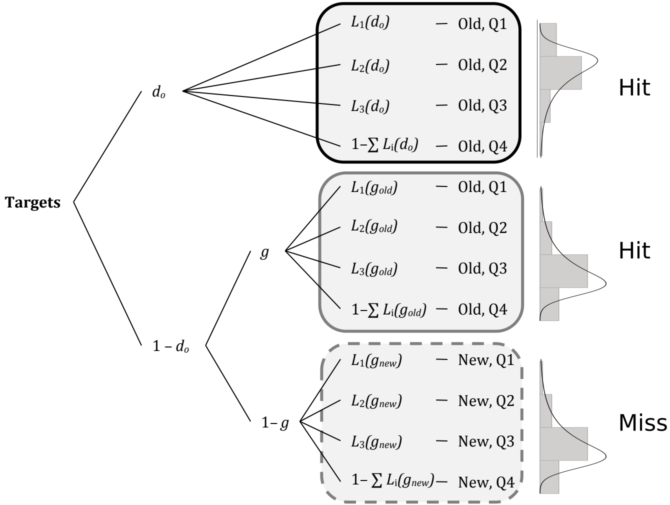
Categorizing RTs

- Depending on the research question and the number of observations, we can use more or less bins
- Note that the bin probabilities always must sum to one!
  - Hence, for 2 bins, we need 1 L-parameter
  - Hence, for 8 bins, we need 7 L-parameters



The RT-extension results in a new (larger) MPT model

- Each set of L parameters represents a histogram for *one* latent RT distribution



Using Histogram-MPTs in Practice

1. Categorize continuous variable into discrete bins
2. Derive constraints which of the latent component distributions are identical
   - Example: Identical RT distribution of "guessing old" for targets and lures
3. Fit the new RT-MPT model
   - Data: Frequencies for all combinations of discrete responses and RT bins

MPT category	RT bins			
	Very fast	Fast	Slow	Very Slow
Target: Hit	44	36	15	4
Target: Miss	15	8	13	23
Lure: False alarm	4	17	22	19
Lure: Correct rejection	31	41	9	4

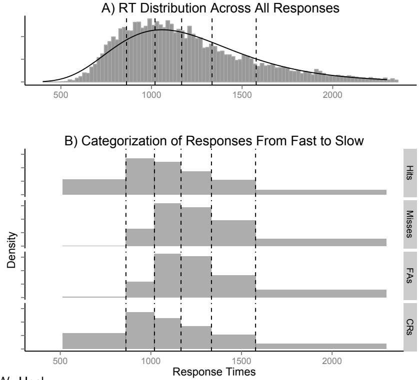
Details

- Concerning (1): How to define RT boundaries for the bins?
- Concerning (2): Is the new model identifiable?

Problem 1: Which RT Boundaries?

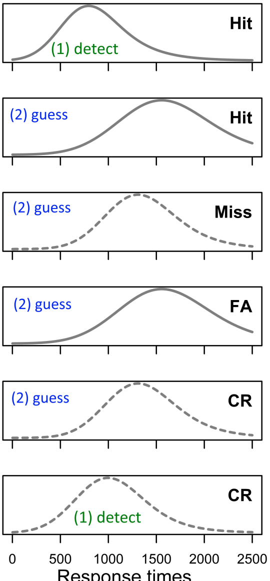
A Principled Strategy to Categorize RTs (details: Heck & Erdfelder, 2016)

- Compute separate RT bounds per participant (individual differences)
  - Interpretation: "Are responses fast or slow *relative* to the overall speed of responding of a person?"
- With 2 RT bins:
  1) Compute the geometric mean across *all* RTs: `bound = exp(mean(log(RTs)))`
  2) Categorize responses as "fast" or "slow"



Identifiability of latent RT distributions

- The number of latent RT distributions must be equal or smaller than the number of observed distributions
  - 2HTM: Maximum of 4 latent RT distributions (observed: hit, miss, FA, CR)
- Some latent distributions directly result in observable distributions
  - These latent distributions are directly identifiable
  - 2HTM: Misses and FAs are always guessing RTs!
- A stepwise procedure allows to check the identifiability of the remaining component distributions
  - cf. Appendix



MPT-RT: Summary

Recipe for MPT-RTs in practice

1. Categorize continuous variable into discrete bins
   - Example: RTs faster or slower than geometric mean?
2. Derive constraints which of the latent component distributions are identical
3. Check identifiability (and revise model)
4. Collect data with RTs
5. Fit the new MPT-RT model
6. Test hypotheses about the relative speed of processes (L parameters)
   - Very simple for 2 RT bins: one L parameter per process (“fast” vs. “slow”)
   - Are processes equally fast? (equality constraints: $L_f = L_s$)
   - Is process i faster than process j? (order constraints: $L_f > L_s$)

Summary

Advantages of the Histogram Approach

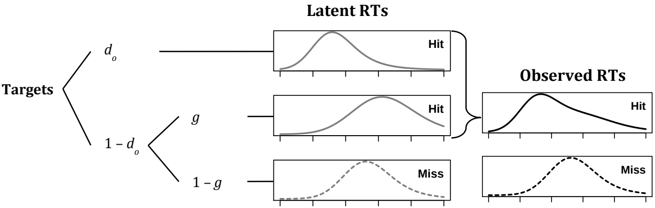
- Does not require parametric assumptions for latent distributions
- Simple: one can use standard MPT software
  - multiTree, TreeBUGS
- Allows novel tests of theories by including RTs
  - Recognition heuristic (Heck & Erdfelder, 2017)

Generalized Processing Tree Models

- Heck, D. W., Erdfelder, E., & Kieslich, P. J. (2018). Generalized processing tree models: Jointly modeling discrete and continuous variables. *Psychometrika.*

Generalized processing tree (GPT) models

- Main difference: Parametric assumptions for component distributions
- The type of distribution depends on continuous variable
  - RTs: log-normal, ex-Gaussian, ...
  - Mouse-tracking measures (see below): Normal distribution
  - Neuro-psychological measures: ...
- The distributions are described by parameters $\boldsymbol{\eta}$
  - Normal distribution: mean and SD
  - ex-Gaussian: mean, SD, and mean of exponential

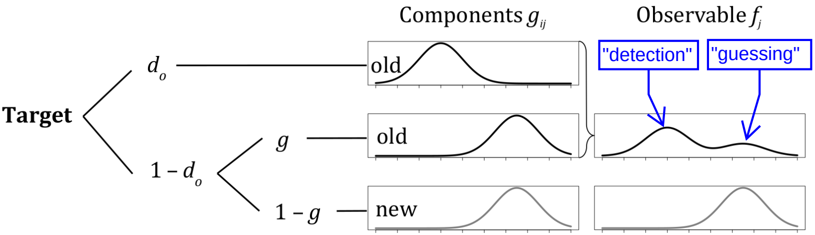


Benefits of the GPT Framework

A) Increased precision in estimating MPT parameters $\boldsymbol{\theta}$
B) Unidentifiable MPT models can become identifiable
C) Flexibility and simplicity

Higher precision of MPT-parameter estimates in GPTs

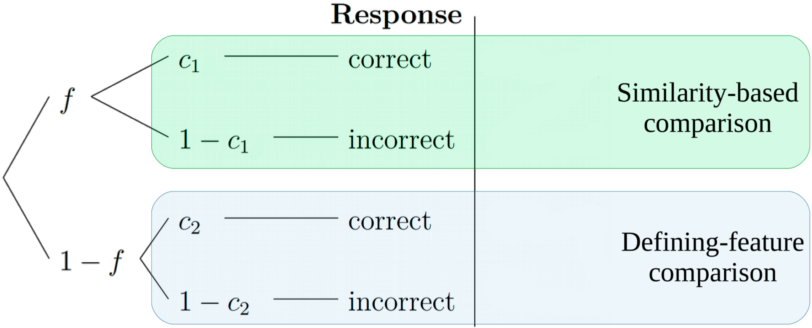
- The more distinct the latent distributions, the smaller the standard error
- Intuition: Continuous variables improve the "classification" which trials belong to which latent processing states
- 2HTM: "Fast RTs are due to detection, slow RTs are due to guessing"



Identifiability of GPT Models

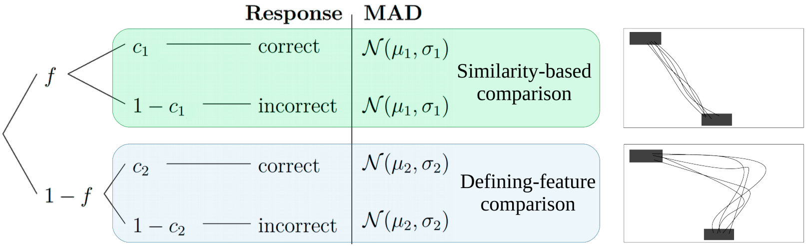
Example: The feature comparison model of semantic categorization

- The theory assumes two different processes
- f = probability of Process 1 (similarity-based comparison)
- c_1 = accuracy of similarity-based comparison
- c_2 = accuracy of defining-feature comparison
- With discrete responses only, the (MPT) model is not identifiable
  - only 1 free category for 3 free parameters



Identifiability of the feature comparison model

- Solution: Assume Gaussian component distributions for continuous variable
  - MAD = maximum absolute deviation in mouse-tracking
- Order constraint for mean parameters: $\mu_c < \mu_d$
  - Interpretation: More direct trajectories/small MADs for similarity-based comparison
- With both discrete *and* continuous data, model is identifiable
  - The different component distribution allow to disentangle the two processes



Flexibility: GPT Model Specification

GPTs as general-purpose measurement models

- GPTs can be specified as easily as MPTs
- Implemented in the R package gpt (https://github.com/danheck/gpt)
- Define model in a text file similar to EQN
- Type of latent distribution(s) defined within R (e.g., `latent="normal"`)

GPT version of 2-high-threshold model

```
# Tree ; Categ. ; MPT equation ; mean, SD (normal distr.)
target ; hit    ;  d           ; m_d,  sig
target ; hit    ; (1-d)*g      ; m_g,  sig
target ; miss   ; (1-d)*(1-g)  ; m_g,  sig

lure   ; cr     ;  d           ; m_d,  sig
lure   ; fa     ; (1-d)*g      ; m_g,  sig
lure   ; cr     ; (1-d)*(1-g)  ; m_g,  sig
```

Illustration of the gpt Package

```r
library("gpt")
# data from 2(response bias) x 2(memory strength) design:
# labels: "o30s_cr" = 30% old items / strong memory / correct rejection
head(heck2016, 3)
```

```
##       cat   rt
## 1 o30s_cr 1123
## 2 o30s_cr  671
## 3 o30s_cr  728
```

```r
modelfile <- "models/2htm_exgauss_2x2.txt"
# first lines:
```

```
## # 30% old / strong memory
## lure_s30;     o30s_cr   ;  (1-dn_s)*(1-g30) ; mu,sig,lambda_g_new30
## lure_s30;     o30s_cr   ;  dn_s             ; mu,sig,lambda_dn_s
## lure_s30;     o30s_fa   ;  (1-dn_s)*g30     ; mu,sig,lambda_g_old30
##
## target_s30;   o30s_hit  ;   do_s            ; mu,sig,lambda_do_s
## target_s30;   o30s_hit  ;  (1-do_s)*g30     ; mu,sig,lambda_g_old30
## target_s30;   o30s_miss ;  (1-do_s)*(1-g30) ; mu,sig,lambda_g_new30
##
## # 30% old / weak memory
## lure_w30;     o30w_cr   ;  (1-dn_w)*(1-g30)  ; mu,sig,lambda_g_new30
## lure_w30;     o30w_cr   ;  dn_w              ; mu,sig,lambda_dn_w
## lure_w30;     o30w_fa   ;  (1-dn_w)*g30      ; mu,sig,lambda_g_old30
```

gpt Package: Model Fitting

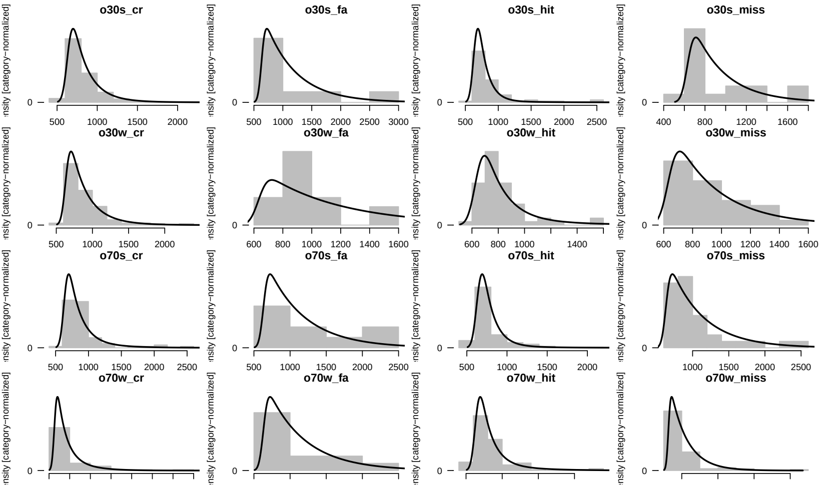
```r
fit <- gpt_fit(x = "cat",          # MPT category
               y = "rt",           # name of continuous variable(s)
               data = heck2016,    # example data for 1 person
               file = modelfile,   # GPT model file
               latent="exgauss",   # family of latent RT distributions
               restrictions=list("dn_s=do_s", "dn_w=do_w"))
```

```r
fit
```

```
##                Estimate     SE CI.lower CI.upper
## dn_s              0.741  0.027    0.688    0.794
## dn_w              0.477  0.033    0.411    0.542
## g30               0.189  0.031    0.128    0.250
## g70               0.260  0.038    0.186    0.335
## lambda_dn_s     172.943 18.251  137.171  208.715
## lambda_dn_w     191.283 34.353  123.952  258.613
## lambda_do_s     128.458 14.744   99.560  157.356
## lambda_do_w     151.936 18.482  115.711  188.161
## lambda_g_new30  311.987 30.776  251.667  372.306
## lambda_g_new70  460.603 36.226  389.601  531.606
## lambda_g_old30  531.304 86.585  361.602  701.007
## lambda_g_old70  517.453 74.722  371.000  663.906
## mu              633.785  5.365  623.269  644.301
## sig              49.165  4.037   41.251   57.078
```

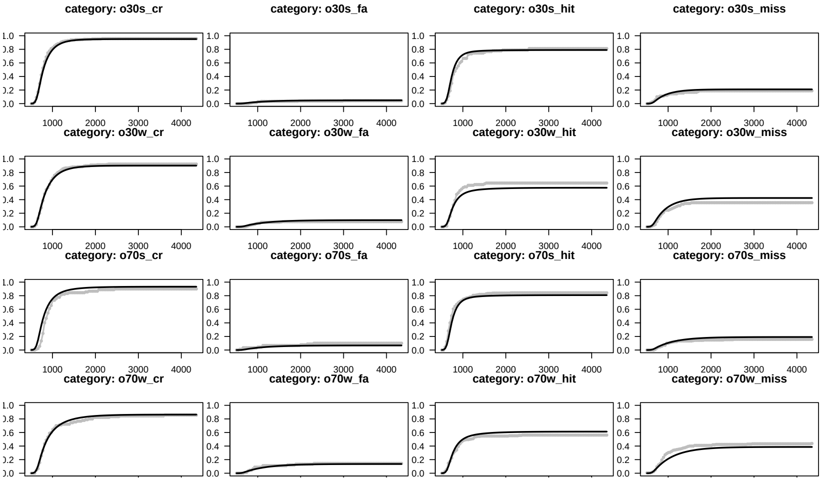
gpt Package: Model Fit

```r
hist(fit)
```



gpt Package: Model Fit

```
plot(fit)   # cumulative densities
```



RT-MPT: Serial-process modeling of RTs

- Klauer, K. C., & Kellen, D. (2018). RT-MPTs: Process models for response-time distributions based on multinomial processing trees with applications to recognition memory. *Journal of Mathematical Psychology, 82*, 111–130.

RT-MPT

RT-MPT Models (Klauer & Kellen, 2018)

- Serial processing assumption: Observed RTs within each MPT branch are the result of a sequence of underlying processes
  1) Time for encoding and response execution
  2) Completion time for each state in the MPT model
  3) Observed RTs in a branch are the sum of encoding and all relevant processing times
- 2HTM components:
  - "Detection RTs": Encoding + Detection
  - "Guessing RTs": Encoding + unsuccessful detection + guessing

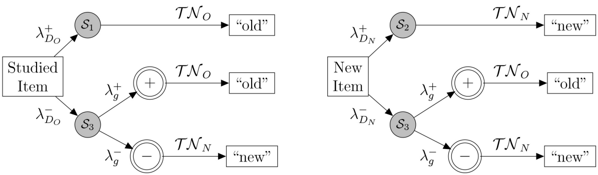
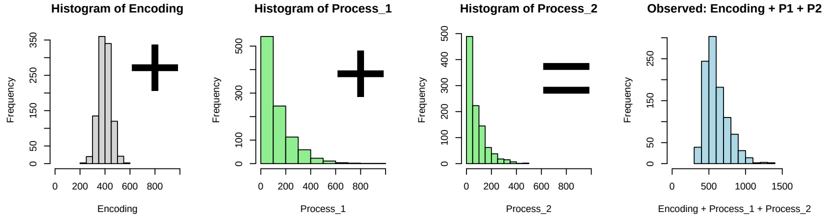


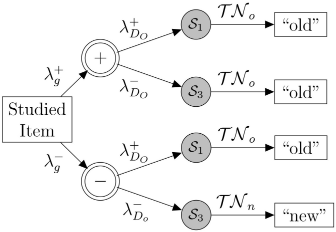
Illustration of the parametric assumptions

- Encoding and response execution (gray): truncated normal distribution (with mean μ and SD σ)
- Completion times (green): exponential distribution (with rate parameters λ)
- Observed time (blue): sum of encoding and processing times
- Note: Encoding and all completion times are independent



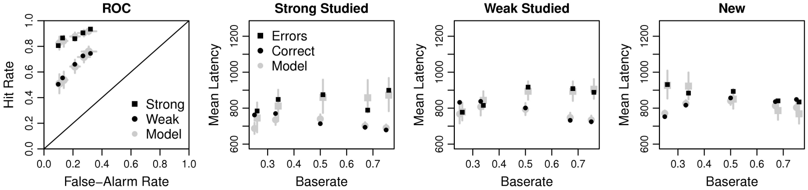
The ordering of the latent MPT states matters

- The 2HTM permits two possible orders:
  - A) "Detect-Guess Model": First detection, then guessing (previous slides)
  - B) "Default-Interventionist Model": First guessing (by default) and the detection process can intervent (see below)
- With the RT-MPT extension, both versions can be tested against each other



Empirical test of different 2HTM versions

- Bayesian hierarchical model (person random effects)
- The Default-Interventionist model fits 5 data sets better
- Effect of manipulations
  - No effect of memory strength on completion time of detection
  - Faster completion times for guessing if response matches the response-bias-condition
- The model fit is satisfactory:



Fitting RT-MPTs

- R package `rtmpt` (Hartmann et al., 2020; Hartmann & Klauer, 2020)
- User-friendly C++ implementation for MCMC sampling
- Example of model specification:

```
# Tree   ;  # Category   ;  # EQN
Target   ;  hit          ;  Do
Target   ;  hit          ;  (1-Do)*g
Target   ;  miss         ;  (1-Do)*(1-g)
Lure     ;  f_a          ;  (1-Dn)*g
Lure     ;  c_r          ;  Dn
Lure     ;  c_r          ;  (1-Dn)*(1-g)

constant:  g = 0.5
suppress:  g-, g+

resp:  Target  ;  hit   ;  0
resp:  Target  ;  miss  ;  1
resp:  Lure    ;  f_a   ;  0
resp:  Lure    ;  c_r   ;  1
```

Summary

Summary & Conclusion

RT-Extended MPT Models (Heck & Erdfelder, 2016)

A) No assumptions about shape of latent distributions
B) RT-extended MPT models are also MPT models
C) Testing relative speed of processes (stochastic dominance)

GPT Models (Heck, Erdfelder, & Kieslich; 2018)

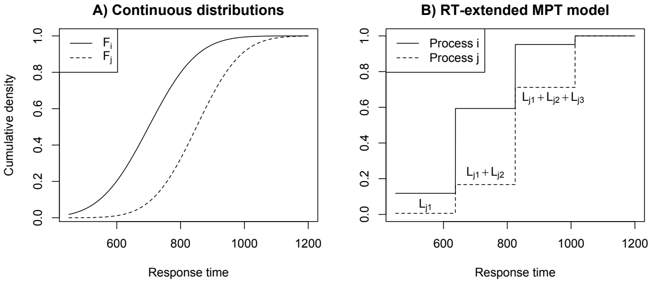
A) General approach for modeling discrete and continuous data
B) New tests of psychological theories (e.g., mouse-tracking)
C) MPT parameters estimated more precisely
D) User-friendly software (R package gpt)

RT-MPT (Klauer & Kellen, 2018)

A) Assumption of serial processing (sum of encoding and completion times)
B) Hierarchical Bayesian

Appendix

- Is process i faster than process j? (order constraints)



Appendix: Stepwise Identifiability

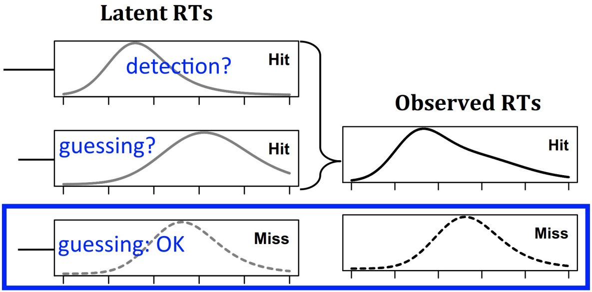
A stepwise procedure to check identifiability

1) Those latent distributions that directly result in observable distributions are directly identifiable
2) Look for 2-component mixtures
3) Is one of latent RT distributions directly identifiable from the first step?
4) It follows that the second RT distribution is also identifiable!
5) Look for 3-component mixtures
6) Check whether 2 of the 3 components are identified
7) . . .

Details: Next slides

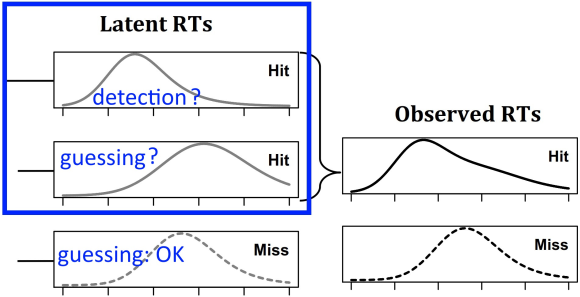
Step 1:
Observable latent RT distributions

2HTM: guessing RTs = Miss RTs



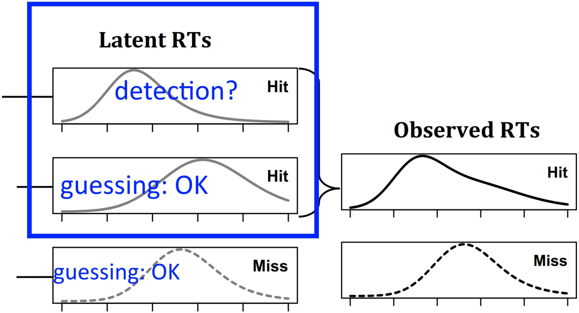
Step 2:
Check 2-component mixtures

2HTM: Hit RTs



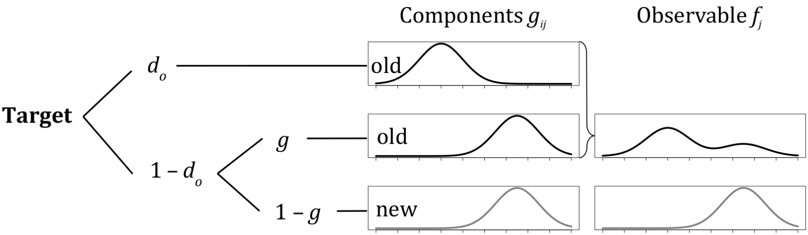
Step 3: Check whether components are identifiable

2HTM: detection RT identifiable



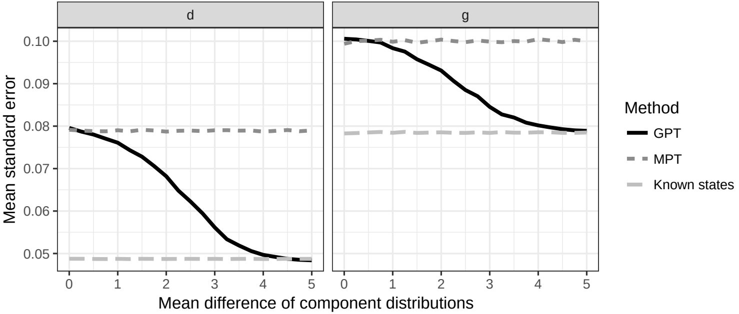
Appendix: GPT Estimates are More Precise

- Higher precision of parameter estimates $\hat{\theta}$
  - The more distinct the latent distributions, the smaller the standard error of $\hat{\theta}$
  - Intuition: Continuous variables improve the classification of trials to latent processing states
  - Upper bound: Precision of MPT model
  - Lower bound: Latent states known



Appendix: GPT Estimates are More Precise

- Simulation of the 2HTM with Gaussian component distributions
  - $\mu^{\text{detect}} = 0$ vs. $\mu^{\text{guess}} = 0, \ldots, 5$
- Results
  - More distinct distributions: smaller standard error of $\hat{\boldsymbol{\theta}}$
  - Upper bound: Precision of MPT model
  - Lower bound: Latent states known



Appendix: Formal Definition of GPT Models

For a vector of discrete responses $\boldsymbol{x}$, a matrix of continuous variables $\boldsymbol{Y}$:

- The **joint distribution** f is a finite mixture
  - For a discrete response x and continuous response(s) $\boldsymbol{y}$:

$$f(x, \boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\eta}) = \sum_{j=1}^{J} \delta_{C_j}(\{x\}) \sum_{i=1}^{I_j} p_{ij}(\boldsymbol{\theta}) g_{ij}(\boldsymbol{y} \mid \boldsymbol{\eta})$$

- **Mixture weights** $p_{ij}(\boldsymbol{\theta})$
  - Identical to MPT-branch probabilities (probability parameters $\boldsymbol{\theta}$)

$$p_{ij}(\boldsymbol{\theta}) = c_{ij} \prod_{s=1}^{S} \theta_s^{a_{ijs}} (1 - \theta_s)^{b_{ijs}}$$

- **Basis distributions** $g_{ij}(\boldsymbol{y} \mid \boldsymbol{\eta})$ for latent states
  - E.g., normal, exGaussian, exWald,... distributions
  - Product-distributions for multivariate continuous data

Appendix: Formal Definition of GPT Models

- GPT models are a set of parameterized distributions:

$$\mathcal{M}^{\mathsf{GPT}}(\Theta = [0,1]^{S_1}, \Lambda \subset \mathbb{R}^{S_2}) = \{f(x, \boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\eta}) \mid \boldsymbol{\theta} \in \Theta, \boldsymbol{\eta} \in \Lambda\}$$

- Likelihood where trials $k = 1, \ldots, K$ fall into disjoint sets M_t that are modeled by $t = 1, \ldots, T$ processing trees

$$L(\boldsymbol{\theta}, \boldsymbol{\eta} \mid \boldsymbol{x}, \boldsymbol{Y}) = \prod_{t=1}^{T} \prod_{k \in M_t} f_t(x_k, \boldsymbol{y}_k \mid \boldsymbol{\theta}, \boldsymbol{\eta})$$

Expectation-Maximization (EM) algorithm

- E: Compute expected probabilities $\boldsymbol{z}$ of being in the cognitive states
  - Continuous variables inform the state-vector $\boldsymbol{z}$
- M: Maximize likelihood of continuous parameters given the latent-states $\boldsymbol{z}$

Illustration

- E-step estimates the probability to be in state i in trial k:

$$P(z_k = i \mid \boldsymbol{\theta}, \boldsymbol{\eta}, x_k, \boldsymbol{y}_k) = \frac{P(z_k = i \mid \boldsymbol{\theta}, x_k) g_{ij}(\boldsymbol{y}_k \mid \boldsymbol{\eta})}{\sum_i P(z_k = i \mid \boldsymbol{\theta}, x_k) g_{ij}(\boldsymbol{y}_k \mid \boldsymbol{\eta})}$$

Cat.	RT [ms]	Conf. [1-10]	ERP [mV]	$z_k = 1$	...	$z_k = I$
c_1	551	3	1.324	0.43	...	0.09
c_1	502	1	0.921	0.19	...	0.56
c_2	470	6	2.231	0.30	...	0.00
c_1	733	4	1.010	0.14	...	0.47

Appendix: Identifiability

Identifiability: GPT models with identifiable MPT structure are identifiable if (cf. distribution-free approach):

- Component distributions are observable (Yantis et al., 1991)
- Stepwise deletion of identifiable component distributions (Heck & Erdfelder, 2016)
- Specific matrix has full rank (Heck & Erdfelder, 2016)

Alternative strategy

- Using order constraints to identify GPT models with a nonidentifiable MPT structure
- Label switching of processing paths with component distributions